

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A method for routing a transaction to a front-end server, comprising:

identifying at least one attribute-based category for said transaction;

attempting to identify at least one of a plurality of front-end servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category; and

when at least one of the front-end servers is identified, routing said transaction to one of said at least one identified front-end server;

when no front-end server is identified, routing said transaction to a default one of the front-end servers; and

determining whether the transaction is associated with a new attribute-based category, and if so, assigning the new attribute based category to one of the front end servers;

determining a status of an attribute-based category; and

deallocating said attribute-based category from said front-end server to which it is assigned when said status is inactive.

Claim 2 (original): A method as in claim 1, further comprising assigning said at least one attribute-based category to said transaction.

Claim 3 (original): A method as in claim 2, wherein assigning said at least one attribute-based category to said transaction comprises associating a tag with said transaction.

Claim 4 (original): A method as in claim 1, wherein identifying said at least one front-end server comprises comparing said attribute-based category for said transaction to assigned attribute-based categories for said plurality of front-end servers.

Claim 5 (original): A method as in claim 1, further comprising determining whether said at least one front-end server is available for processing said transaction.

Claim 6 (original): A method as in claim 1, further comprising rerouting said transaction to another of said plurality of front-end servers when said identified server refuses said transaction.

Claim 7 (cancelled)

Claim 8 (previously presented): A method as in claim 1, further comprising notifying a workload manager of said at least one front-end server assigned to said new attribute-based category.

Claim 9 (cancelled)

Claim 10 (currently amended): An apparatus for routing a transaction to a front-end server, comprising:

computer readable storage media; and

computer readable program code stored on said computer readable storage media,

comprising:

program code for attempting to identify at least one attribute-based category for said transaction;

program code for identifying at least one of a plurality of front-end servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category; and

program code for, when at least one of the front-end servers is identified, routing said transaction to one of said at least one identified front-end server; and

program code for, when no front-end server is identified,

routing said transaction to a default one of the front-end servers;

determining whether the transaction is associated with a new attribute-based category; and

assigning the new attribute-based category to one of the front-end servers;

program code for assigning a number of attribute-based categories to each of said plurality of front-end servers, wherein said program code for routing said transaction to one of said identified front-end servers routes said transaction according to said assigned attribute-based categories;

program code for determining a status for each of said assigned attribute-based categories; and
program code for deallocating said assigned attribute-based categories when said status thereof is inactive.

Claim 11 (original): An apparatus as in claim 10, further comprising program code for assigning said at least one attribute-based category to said transaction.

Claim 12 (original): An apparatus as in claim 10, wherein said attribute-based category is based on at least one “real” attribute of said transaction.

Claim 13 (original): An apparatus as in claim 10, wherein said attribute-based category is based on at least one “perceived” attribute of said transaction.

Claim 14 (original): An apparatus as in claim 10, further comprising a user table for assigning said at least one attribute-based category to said transaction.

Claim 15 (original): An apparatus as in claim 10, further comprising:

program code for determining whether said at least one identified server is available for processing said transaction; and

program code for rerouting said transaction to another of said plurality of servers when at least one identified server is unavailable for processing said transaction,

Claim 16 (cancelled)

Claim 17 (currently amended): An apparatus as in claim 10[6], wherein said program code for assigning at least one attribute-based category to each of said plurality of servers bases the assignment at least in part on an affinity of transaction attributes.

Claim 18 (currently amended): An apparatus as in claim 10[6], further comprising a workload manager table for recording said assigned attribute-based categories.

Claims 19-22 (cancelled)

Claim 23 (previously presented): A method as in claim 1, wherein identifying said at least one attribute-based category for said transaction comprises identifying a “perceived” attribute of said transaction.

Claim 24 (previously presented): A method as in claim 23, wherein the identified perceived” attribute is the computer originating the transaction.

Claim 25 (previously presented): A method as in claim 23, wherein the identified “perceived” attribute is the user originating the transaction.

Claim 26 (previously presented): A method as in claim 23, wherein the identified “perceived” attribute is a class of users from which the transaction originates.

Claims 27 - 29 (cancelled)

Claim 30 (currently amended): A method as in claim 1, further comprising: ~~A method for routing a transaction to a front-end server, comprising:~~

~~—— identifying at least one attribute-based category for said transaction;~~

~~—— attempting to identify at least one of a plurality of front-end servers to process said~~

~~transaction based at least in part on said identified attribute-based category of said transaction~~

~~and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category;~~

routing said transaction to one of said at least one identified front-end servers; and

one or more of said front-end servers,

maintaining its own table of attribute-based categories for transactions that it has processed;

for each attribute-based category in its table, maintaining an indication of when a transaction corresponding to the attribute-based category was last processed by the front-end server; and

after a predetermined time of not processing a transaction corresponding to an attribute-based category in its table, broadcasting an indication of this event to a plurality of workload managers that can route transactions to the front-end server.

Claim 31 (previously presented): A method as in claim 30, further comprising:

upon a workload manager's receipt of said broadcast indication, the workload manager updating its own table of assignments between attribute-based categories and front-end servers.

Claim 32 (previously presented): An apparatus as in claim 10, further comprising program code to update, in response to broadcast indications from said front-end servers, a table of which attribute-based categories are assigned to which front-end servers, said table being maintained by and for a particular workload manager.

Claim 33 (cancelled)